1. 基本字符测试

输入：–c char.c

期望输出： char.c, 字符数: 31

实际输出： 结果未正确显示到屏幕上

分析：输出结果模块中，错误地将所有的输出都输出到了文件中

结果：已改正，现在可以正确输出结果

1. 不可显示字符测试

输入： -c charwithspace.c

期望输出：charwithspace.c, 字符数: 38

实际输出：charwithspace.c, 字符数: 40

分析：错误地将换行符’\n’也作为字符统计了

结果：已改正，现在可以输出正确结果

1. 单词和行数测试

输入：-w -l wordtest.c

期望输出： wordtest.c, 单词数: 16

wordtest.c, 行数: 4

实际输出： wordtest.c, 单词数: 16

wordtest.c, 行数: 4

分析：符合预期输出

1. 扩展行数测试

输入：-a atest.c

期望输出：atest.c, 代码行/空行/注释行: 10/3/9

实际输出：atest.c, 代码行/空行/注释行: 10/3/9

分析：符合预期输出

1. 停用词测试

输入：-w stoptest.c -e stoplist.txt

期望输出：stoptest.c，单词数：15

实际输出：stoptest.c，单词数：15

分析：符合预期输出

1. 文件夹遍历测试

输入：-s -w -a C:\Users\Star\Desktop\SoftTest\\*.c

期望输出&实际输出：

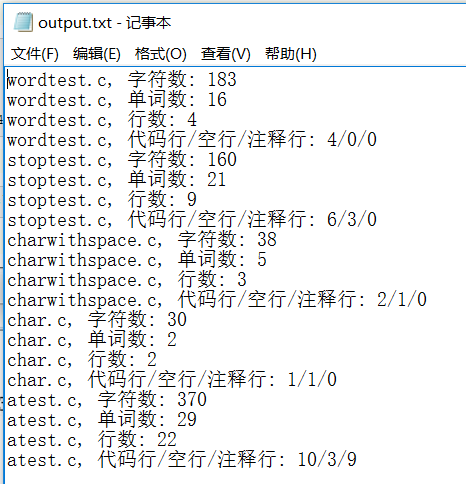


分析：符合预期输出

1. 输出测试：

输入：-s -a -w -c -l C:\Users\Star\Desktop\SoftTest\\*.c -o output.txt

期望输出&实际输出：

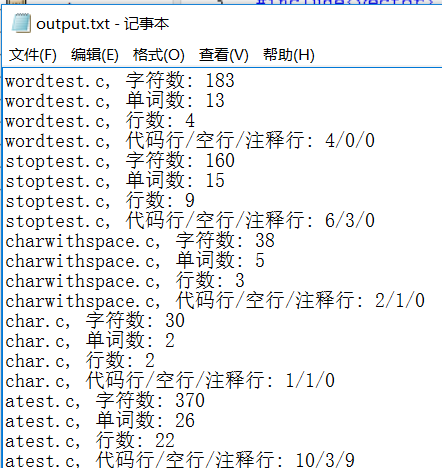


分析：符合预期输出

1. 全套测试：

输入：-s -a -w -c -l C:\Users\Star\Desktop\SoftTest\\*.c -o output.txt -e stoplist.txt

期望输出：



实际输出：目标文件并未出现任何字符

分析：由于指令过长，没有设置足够的数组长度来存储指令，导致解析失败

结果：将指令最大长度设置为150，得到了正确结果。

1. 错误指令测试1

输入：-c -d char.c charwithspace.c

期望输出：不可预知

实际输出：charwithspace.c, 字符数: 38

分析：忽略了错误指令-d，并正确统计了倒数第一个文件名内的字符数，程序健壮性良好

1. 错误指令测试2

输入：-e -c char.c

期望输出：不可预知

实际输出：无输出，但程序没有崩溃

分析：程序将-c识别为停用词表的路径，char.c仍然是待统计文件，由于没有给出需要统计的类型，所以没有进行统计。